

© EPODOC / EPO

PN - JP7134656 A 19950523
PD - 1995-05-23
PR - JP19930281091 19931110
OPD - 1993-11-10
TI - INTERRUPTION CONTROL METHOD AND MULTI-TASK SYSTEM
FOR REALIZING THE SAME
IN - WATANABE HIROYASU; IWABUCHI YOICHI; YAMADA JUNJI;
KIKUCHI KAZUNARI
PA - CANON KK
IC - G06F9/46 ; G06F9/46

© PAJ / JPO

PN - JP7134656 A 19950523
PD - 1995-05-23
AP - JP19930281091 19931110
IN - KIKUCHI KAZUNARI; others03
PA - CANON INC
TI - INTERRUPTION CONTROL METHOD AND MULTI-TASK SYSTEM
FOR REALIZING THE SAME
AB - PURPOSE: To perform control with a processing for which the end
of an interruption processing is written by a return instruction
without requiring a special procedure by confirming whether or not
interruption is multiple interruption and changing a return
destination in the return instruction of the interruption processing to
a task switching processing when it is not the multiple interruption.
- CONSTITUTION: In the interruption processing to which execution is
shifted, by the execution of an interruption return instruction
present at the end of the processing, an address and a state
register value present in a highest order on a stack 723 are read
and the execution is restarted with the state register value. That is,
in the case of the interruption which is not the multiple interruption,
the highest order of the stack 723 is the leading address and the
state register value of the task switching processing present at a
point (d) and a task processing is returned after passing through
the task switching processing. On the other hand, in the case of the
multiple interruption, the highest order of the stack 723 is an
execution address and the state register value before the
interruption present at the point (a) and the interruption processing
for which the interruption is performed is returned.
I - G06F9/46

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-134656

(43) 公開日 平成7年(1995)5月23日

(51) Int.Cl.⁶

G 0 6 F 9/46

識別記号

庁内整理番号

F I

技術表示箇所

3 4 0 B 8120-5B

3 1 0 Z 8120-5B

審査請求 未請求 請求項の数 7 O L (全 9 頁)

(21) 出願番号 特願平5-281091

(22) 出願日 平成5年(1993)11月10日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 菊地 一成

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(72) 発明者 岩淵 洋一

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(72) 発明者 渡辺 浩康

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(74) 代理人 弁理士 大塚 康徳 (外1名)

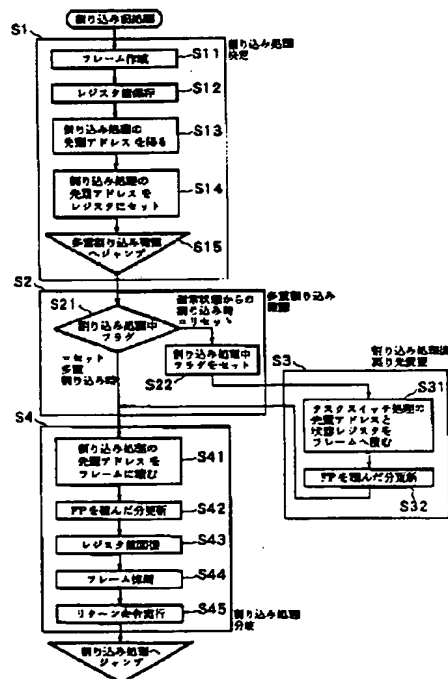
最終頁に続く

(54) 【発明の名称】 割り込み制御方法及びそれを実現するマルチタスクシステム

(57) 【要約】

【目的】 割り込み処理の最後でタスクスイッチのための特殊な手続きを必要とすることのない、割り込み処理の最後が割り込みリターン命令で書かれている割り込み処理で、リアルタイム処理向けマルチタスクシステムの割り込み制御が可能な割り込み制御方法を提供することを目的とする。

【構成】 多重割り込みか否かの確認をし (S2)、多重割り込みでない場合に、割り込みリターン命令での戻り先を変更する割り込み処理後の戻り先をタスクスイッチ処理に変更する (S3) ことにより、割り込みリターン命令でのタスクスイッチ処理の実行が実現される。この戻り先の変更は、特に、割り込み受付時、該割り込みに対応した処理の実行に先立ち処理される割り込み前処理で行われる。



【特許請求の範囲】

【請求項1】 リアルタイム処理が要求されるマルチタスクシステムにおける割り込み制御方法であって、割り込みが多重割り込みか否かを確認し、多重割り込みでない場合に、割り込み処理のリターン命令での戻り先をタスクスイッチ処理に変更することを特徴とする割り込み制御方法。

【請求項2】 前記タスクスイッチ処理への変更は、リターン命令で参照するスタック上の戻り先アドレスをタスクスイッチ処理の先頭アドレスへ変更する処理を含むことを特徴とする請求項1記載の割り込み制御方法。

【請求項3】 前記タスクスイッチ処理への変更は、割り込み受付時に該割り込みに対応した処理の実行に先立ち実行されることを特徴とする請求項1または2記載の割り込み制御方法。

【請求項4】 リアルタイム処理が要求されるマルチタスクシステムにおいて、割り込みが多重割り込みか否かを確認する確認手段と、多重割り込みでない場合に、割り込み処理のリターン命令での戻り先を変更する変更手段とを備えることを特徴とするマルチタスクシステム。

【請求項5】 前記変更手段は、リターン命令で参照するスタック上の戻り先アドレスを所望の処理の先頭アドレスへ変更するアドレス変更手段を含むことを特徴とする請求項4記載のマルチタスクシステム。

【請求項6】 前記アドレス変更手段は、スタック上で元の戻り先アドレスの上に所望の処理の先頭アドレスを積み上げるスタック変更手段と、スタックポインタを所望の処理の先頭アドレスに変更するスタックポインタ変更手段とを備えることを特徴とする請求項5記載のマルチタスクシステム。

【請求項7】 前記確認手段及び変更手段は、割り込み受付時に該割り込みに対応した処理の実行に先立ち動作することを特徴とする請求項4乃至6のいずれか1つに記載のマルチタスクシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、リアルタイム処理が要求されるマルチタスクシステムの割り込み制御方法それを実現するマルチタスクシステムに関するものである。

【0002】

【従来の技術】 リアルタイム処理が要求されるコンピュータシステムにおいては、外界の処理（以下、外部処理）と同期しながら、プログラムの実行（以下、内部処理）を行う必要がある。このため、外部処理において許容されている時間内に、これに対応した内部処理を完了させなければならない。この時間的要件を満足させるためには、外部処理が、現在実行されている内部処理よりも時間的要求の厳しい新たな内部処理の実行を要求した時に、現在の内部処理を一旦中断して新に要求された内部

処理を先行して行わなければならない。

【0003】 従って、このようなリアルタイム処理をコンピュータで可能とするためには、外部処理からのコンピュータに対する内部処理の実行要求を伝達するために、コンピュータの割り込み処理機構を用いる。また、時間的要件の厳しい内部処理を優先して行うための方法として、必要な内部処理をタスクとしてプログラムしてその各タスク毎に優先度を設け、優先度の高いタスク順にコンピュータのCPU資源を割り当てるプリエンティブな優先度スケジューリングの手法に基づいて、タスクスイッチを行うマルチタスク制御プログラムを利用している。

【0004】 マルチタスク制御プログラムにおいては、タスクの生成、中断、再開、終了や優先度変更などといった各種の機能は、通常サブルーチンコールやソフトウェア割り込みの形式（以下、システムコール）で提供されている。タスクや割り込み処理内でこれらと呼ぶことによって、各機能が実行される。通常、タスクからシステムコールが実行され、その結果他タスクへのタスクスイッチが必要となった場合は、そのシステムコール実行の終了段階でタスクスイッチ処理が実行される。

【0005】 しかし、タスクと同様に割り込み処理中のシステムコールでタスクスイッチ処理を行なった場合は、再度のタスクスイッチの発生によってこの割り込み処理に戻ってくるまで、この割り込み処理の残りの部分の実行が待たされることになったり、また最悪の場合はこれ以降この割り込み処理へ戻るタスクスイッチが発生せず、割り込み処理の実行が継続されない可能性もある。さらに、この割り込み処理が終了しない間は、この割り込み要求レベルと同等かそれより低いレベルの割り込みをコンピュータが受け付けなくなってしまう、外部処理からの他の要求を取りこぼしてしまう可能性もある。

【0006】 このような問題を避けるため、従来のリアルタイム処理が必要なマルチタスクシステムにおいて、割り込み処理中にシステムコールが実行された場合には、その時点でタスクスイッチの必要性があっても行わず、この割り込み処理の最後で呼ばれるタスクスイッチ処理のためのシステムコールの実行時点まで、タスクスイッチが禁止されるようになっている。

【0007】 図3に、リアルタイム処理が要求されるマルチタスクシステムの多重割り込みのタイムチャートを示す。図のように割り込み処理途上で、更に高いレベルの割り込みが発生する多重割り込みの場合には、高レベルの割り込み処理を完了させた後、タスクへ戻る最も低レベル割り込み処理の最後で、タスクスイッチのためのシステムコールによってタスクスイッチが実行される必要がある。このように、割り込み処理内でシステムコールを実行するプログラムは、その終了時に必ずマルチタスク制御プログラムに制御を戻し、タスクスイッチを行

3

う特殊な手続きが不可欠となっている。

【0008】

【発明が解決しようとする課題】しかしながら、リアルタイム処理が要求されるマルチタスクシステムでは、上述のように割り込み処理の最後でタスクスイッチのための特殊な手続きが必要とされていたので、図2のように処理の最後が割り込みリターン命令で書かれている一般的な形式の割り込み処理を用いて、割り込み制御が出来ないという問題があった。

【0009】本発明は、前記従来の欠点を除去し、リアルタイム処理が要求されるマルチタスクシステムにおいて、割り込み処理の最後でタスクスイッチのための特殊な手続きを必要とすることなく、割り込み処理の最後が割り込みリターン命令で書かれている割り込み処理で、リアルタイム処理向けマルチタスクシステムの割り込み制御が可能な割り込み制御方法それを実現するマルチタスクシステムを提供する。

【0010】

【課題を解決するための手段】この課題を解決するために、本発明の割り込み制御方法は、リアルタイム処理が要求されるマルチタスクシステムにおける割り込み制御方法であって、割り込みが多重割り込みか否かを確認し、多重割り込みでない場合に、割り込み処理のリターン命令での戻り先をタスクスイッチ処理に変更することを特徴とする。

【0011】ここで、前記タスクスイッチ処理への変更は、リターン命令で参照するスタック上の戻り先アドレスをタスクスイッチ処理の先頭アドレスへ変更する処理を含む。また、前記タスクスイッチ処理への変更は、割り込み受付時に該割り込みに対応した処理の実行に先立ち実行される。

【0012】また、本発明のマルチタスクシステムは、リアルタイム処理が要求されるマルチタスクシステムにおいて、割り込みが多重割り込みか否かを確認する確認手段と、多重割り込みでない場合に、割り込み処理のリターン命令での戻り先を変更する変更手段とを備えることを特徴とする。

【0013】ここで、前記変更手段は、リターン命令で参照するスタック上の戻り先アドレスを所望の処理の先頭アドレスへ変更するアドレス変更手段を含む。また、前記アドレス変更手段は、スタック上で元の戻り先アドレスの上に所望の処理の先頭アドレスを積み上げるスタック変更手段と、スタックポインタを所望の処理の先頭アドレスに変更するスタックポインタ変更手段とを備える。また、前記確認手段及び変更手段は、割り込み受付時に該割り込みに対応した処理の実行に先立ち動作する。

【0014】

【実施例】

（実施例1）以下、本発明の割り込み制御方法の一実施

4

例を、図1の割り込み前処理のフローチャートと、図7のシステム構成図にしたがって詳細に説明する。

【0015】図7のシステム構成図のコンピュータは、割り込み要求信号線7.3から割り込みを受け付けると、通常処理を中断し再開に必要な現在の実行アドレスと状態レジスタ値とをスタック領域（以下スタック）7.2.3へ保存し、I/Oからバス7.4を通して入力した割り込みベクタ番号（以下ベクタ番号）をキーとしてメモリ内の特定領域に存在する割り込みベクタ表7.2.1（以下ベクタ表）を参照し、ベクタ番号に対応した割り込み処理の先頭アドレス（図7では、割り込み前処理の先頭アドレス）を獲得した後、このアドレスへ実行を移す、といった順で割り込み処理へ移行する。このため、ベクタ表7.2.1には、プログラムの実行以前にそのハードウェアの要件に従った形式で、各ベクタ番号に対応した割り込み処理の先頭アドレスが登録されている。ここでは、割り込み処理の実行に先立って割り込み前処理を実行するので、予め割り込み前処理の先頭アドレスをベクタ表7.2.1へ登録する。

【0016】さて、割り込みが発生すると、上記の手順によって図1に示された割り込み前処理へ実行が移る。この段階では、コンピュータが通常ワークスペースとして使用する図7の先入れ後だしメモリ領域であるスタック7.2.3に、割り込み処理が完了した後に割り込みリターン命令で割り込み以前のプログラムへ復帰するのに最低限必要な、割り込み前のプログラムカウンタ（以下PC）7.1.2の値である実行アドレスと状態レジスタ7.1.4の値とが保存されている。図6に実行中のスタックの様子が示されている。この段階では図6のa点までスタックが使用されている。コンピュータ7.1（CPU）は、スタックの使用済領域を管理するため、図7のスタックポインタ（以下SP）7.1.3を有しており、この段階のSPは点aのアドレスを指示している。

【0017】割り込み前処理では、まず以下に説明する実行すべき割り込み処理の決定S1を実行する（図1参照）。この処理は以下の手順で行われる。

【0018】（割り込み処理の決定S1）割り込み前処理で必要なワークスペースを確保するため、スタック7.2.3内のフレームの作成手続きS1.1を行う。この処理の結果としては、図6のスタック7.2.3上のa点からb点までがフレーム6.1として確保される。フレーム確保により、SP7.1.3はフレームの上端であるb点へ移動する。また、SP7.1.3と同様にフレーム内の使用済領域を管理するためフレームポインタ（以下FP）6.2を作成する。まず、FP6.2として使用されるレジスタ7.1.1の現在の値を保存するため、b点より上の保存レジスタ値領域へその値を積み、その後FP6.2に割り当てられたレジスタはフレームの下端a点をポイントするように初期化される。

【0019】図1のレジスタ値保存手続きS1.2では、

割り込み前処理で使用する他の全てのレジスタ711の値を、図6の保存レジスタ値領域へ順に積む。

【0020】割り込みの先頭アドレスの獲得手続きS13は、割り込み処理の先頭アドレスを、プログラムカウンタ相対アドレッシングなどにより、現在の実行アドレスと実行オブジェクトのリンク時に得られた割り込み処理までのオフセットアドレス値を加算して得る。

【0021】割り込み処理の先頭アドレス保存手続きS14では、割り込み前処理から割り込み処理へ実行を移すまで、得られた割り込み処理の先頭アドレスをレジスタ等に一時保存しておく。

【0022】以上のベクタ番号獲得後の個別処理を実行した後、ジャンプ処理S15で多重割り込み確認S2へジャンプする。これ以下の処理は、ベクタ番号に依存しない共通手続きとして記述可能であり、プログラム領域として図7に722で示すようなメモリ上の配置が可能となる。

【0023】(多重割り込み確認S2) 図1に示すように、多重割り込み確認S2では、多重割り込みの場合とそうでない場合の割り込み処理終了時のリターン先の切り替えを行う。多重割り込みフラグ確認手続きS21では、割り込み処理中に発生した多重割り込みかどうかを判断するため、割り込み処理中フラグを確認する。

【0024】もし、割り込み処理中フラグがセットされていなければ、この割り込みは多重割り込みでないことがわかる。この場合は、この割り込み処理途上で発生する多重割り込みを検出するため、割り込み処理フラグのセット手続きS22を行い、更に図2のように、記述された割り込み処理の最後のリターン命令でタスクスイッチ処理へ実行を移すため、割り込み処理後の戻り先変更S3へ分岐する。

【0025】割り込み処理中フラグが既にセットされていれば、多重割り込み処理分岐S4へ分岐する。本手順以下は、必要に応じてコンピュータを多重割り込み可能状態とすることが出来る。

【0026】(割り込み処理後の戻り先変更S3) 図1の割り込み処理後の戻り先変更S3では、図2のような割り込み処理の最後のリターン命令でタスクスイッチ処理へ実行を移すように、戻り先の変更処理を実行する。タスクスイッチ処理の先頭アドレスと状態レジスタとをフレームに積む手続きS31は、割り込みリターン命令の戻り先をFP62が示している図6のa点の上に積む。このとき、割り込み前のタスクの実行アドレスと状態レジスタとをスタックに残したままにしておくのは、タスクスイッチ処理によって再び割り込まれたタスクへ戻れるようにするためである。この処理のあとフレームを先入れ後だしメモリとして使用するため、FP62の値をd点へ移す手続きS32を実行する。本手順の実行の後、割り込み処理分岐S4を実行する。

【0027】(割り込み処理分岐S4) 割り込み処理分

岐S4は、実行すべき割り込み処理へ分岐するための処理を実行する。割り込み処理の先頭アドレスをフレームに積む手続きS41は、すでに先頭割り込み処理決定S1で求め、かつ一時的にレジスタ等に保存した割り込み処理の先頭アドレスを、図6のFP62が示しているフレームのd点の上に積む。

【0028】この処理のあとフレームを先入れ後だしメモリとして使用するため、FP62の値をe点へ移す手続きS42を実行する。

【0029】レジスタ値回復手続きS43では、図2の割り込み処理が仮想的に割り込みの直後に実行されたように見せるため、SP713とFP62以外の割り込み前処理で使用された全てのレジスタ値を元に戻す。

【0030】フレーム抹消手続きS44では、現在FP62の値が示している図6のe点へSP713を移動する。その後FP62は不要になるので、FP62に使用したレジスタの値を図6の保存レジスタ値領域から回復する。最後のリターン命令の実行S45によって、スタックから割り込みの処理の先頭アドレスが取り込まれ、割り込み処理へ実行が移される。

【0031】さて、実行が移された図2の割り込み処理では、この処理の最後にある割り込みリターン命令の実行によって、スタック上の最上位にあるアドレスと状態レジスタ値とが読み込まれ、この状態レジスタの値で実行が再開される。つまり、多重割り込みでない割り込みの場合は、スタックの最上位にあるのは、図6のd点にあるタスクスイッチ処理の先頭アドレスと状態レジスタ値とであり、タスクスイッチ処理を一旦経由した後、タスク処理へ戻ることになる。タスクスイッチ処理では、タスクスイッチの実行と共に割り込み処理中フラグがセットされる。

【0032】一方、多重割り込みの場合は、スタックの最上位にあるのは、図6のa点にある割り込み前の実行アドレスと状態レジスタ値とであり、割り込みをした低レベル割り込み処理へ戻る。

【0033】図4に、本実施例による以上の多重割り込みのタイムチャートを示す。

【0034】(実施例2) 本発明の割り込み制御方法の他の実施例を図5にしたがって説明する。

【0035】割り込み前処理の実行の後に割り込み処理へ実行を移すため、割り込み処理の先頭アドレス(割り込みベクタ表と同ような構造)の表をメモリ内に別途作成しておく。

【0036】図1に示した実施例1では、実行すべき割り込み処理の決定S1の結果として、実行すべき割り込み処理の先頭アドレスを求めたが、本実施例の図5では、実行すべき割り込み処理の決定S51では、手続きS513で得た直値として埋めこまれている割り込みベクタ(割り込み処理)毎の識別番号を、手続きS514でレジスタやメモリへ一旦保存する。

【0037】後の割り込み処理分岐S54では、この識別番号を使用し手続きS541で、割り込み処理の先頭アドレスを表を参照して求める。この手続きとしては、先に述べた割り込み処理の先頭アドレスをメモリ内に順に並べた表を作成し、その表の先頭アドレスと識別番号から得られるオフセットアドレスとを加算し、その加算結果のアドレスより割り込み処理の先頭アドレスを読み出す。その後、ジャンプ命令S544で割り込み処理へ実行を移す。

【0038】図1の実施例1の多重割り込み確認S2では、割り込み処理中フラグを使用しソフトウェア的に多重割り込みの判断をした。図5の多重割り込み確認手段S52では、コンピュータ内に多重割り込みの発生を確認出来る割り込みコントローラ等のハードウェアがある場合の実施例である。まず、割り込みコントローラのステータスを読み(手続きS521)、このステータスから現在の割り込みよりレベルの低い割り込みが待たされているかどうか手続きS522で確認する。もし、存在しなければ多重割り込みでない割り込みであるので、実施例1と同様の割り込み処理後の戻り先変更S3を実行する。

【0039】尚、本実施例では、最初の低レベルの割り込み処理に入る前に、割り込み前処理でタスクスイッチ処理へのポインタと状態レジスタとをスタックに積んだが、最後の低レベルの割り込み処理からのリターン処理の前に、同様のリターン前処理でタスクスイッチ処理へのポインタと状態レジスタとをスタックに積んでも、本実施例と同様の効果を達成できる。また、本実施例で示した割り込み前処理のフローチャートはその一例であり、同じ機能を達成できればこれに限定されない。例えば、多重割り込み確認を最初に実行し、多重割り込み時にはレジスタ等の保存/回復を最小限にすれば、多重割り込み時の処理の迅速化が図れる。また、本実施例の割り込み前処理は、ソフトウェアにより実現されてもよいし、ハードウェアにより実現されてもよい。更に、ハードウェアとソフトウェアとを組合わせたファームウェア

による実現も出来る。さらに、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器から成る装置に適用しても良い。また、本発明はシステム或は装置にプログラムを供給することによって達成される場合にも適用できることは言うまでもない。

【0040】

【発明の効果】以上説明したように、本発明は、リアルタイム処理が要求されるマルチタスクシステムにおいて、割り込み処理の最後でタスクスイッチのための特殊な手続きを必要とすることなく、割り込み処理の最後が割り込みリターン命令で書かれている割り込み処理で、リアルタイム処理向けマルチタスクシステムの割り込み制御が可能となった。

【0041】また、最近では高級言語によって簡単に割り込みプログラムが記述出来るようになっている。これは、コンパイラが割り込み処理であると宣言されたサブルーチンを、図2の形式へコンパイルすることによる。したがって、高級言語で作成された割り込み処理プログラムも、本発明を適用することによってリアルタイム処理が要求されるマルチタスクシステム下で使用可能となる。

【図面の簡単な説明】

【図1】本発明の実施例1の割り込み前処理の手順を示すフローチャートである。

【図2】本実施例で使用される割り込み処理の手順を示すフローチャートである。

【図3】リアルタイム処理が要求されるマルチタスクシステムの多重割り込みのタイムチャートである。

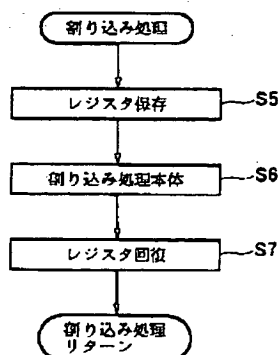
【図4】本実施例による多重割り込みを示すタイムチャートである。

【図5】実施例2に示す割り込み前処理の手順を示すフローチャートである。

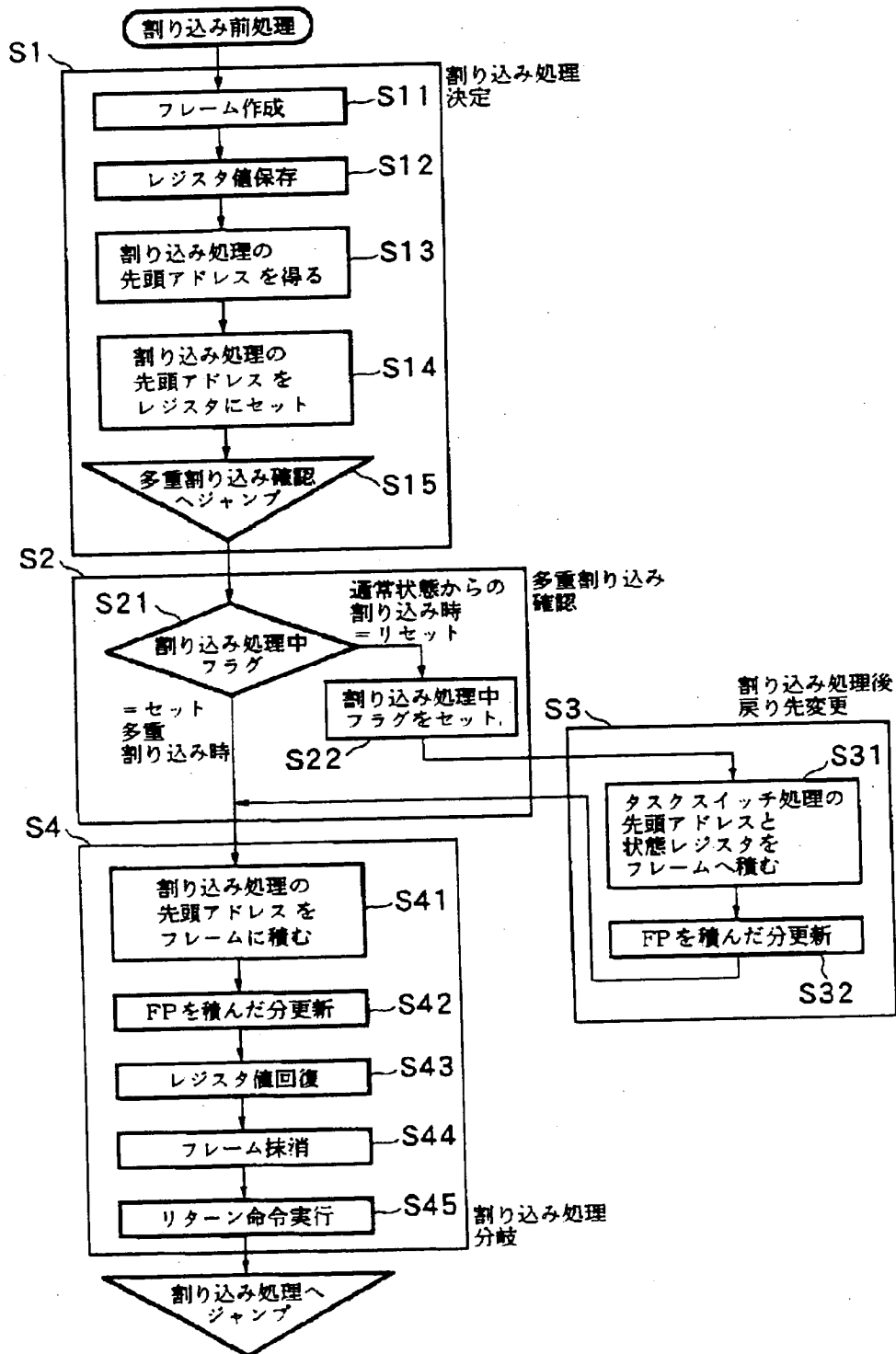
【図6】図1のスタックの使用状況を説明する図である。

【図7】本実施例のシステムを説明するための図である。

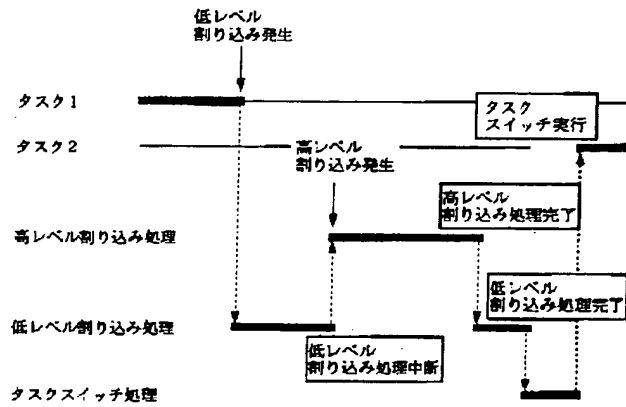
【図2】



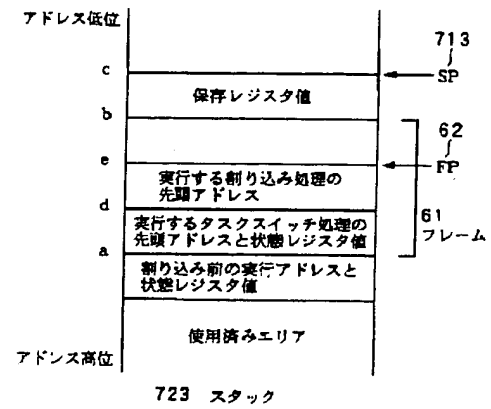
【図1】



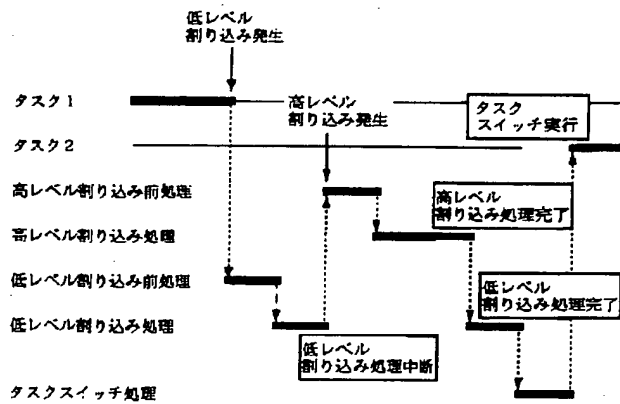
【図3】



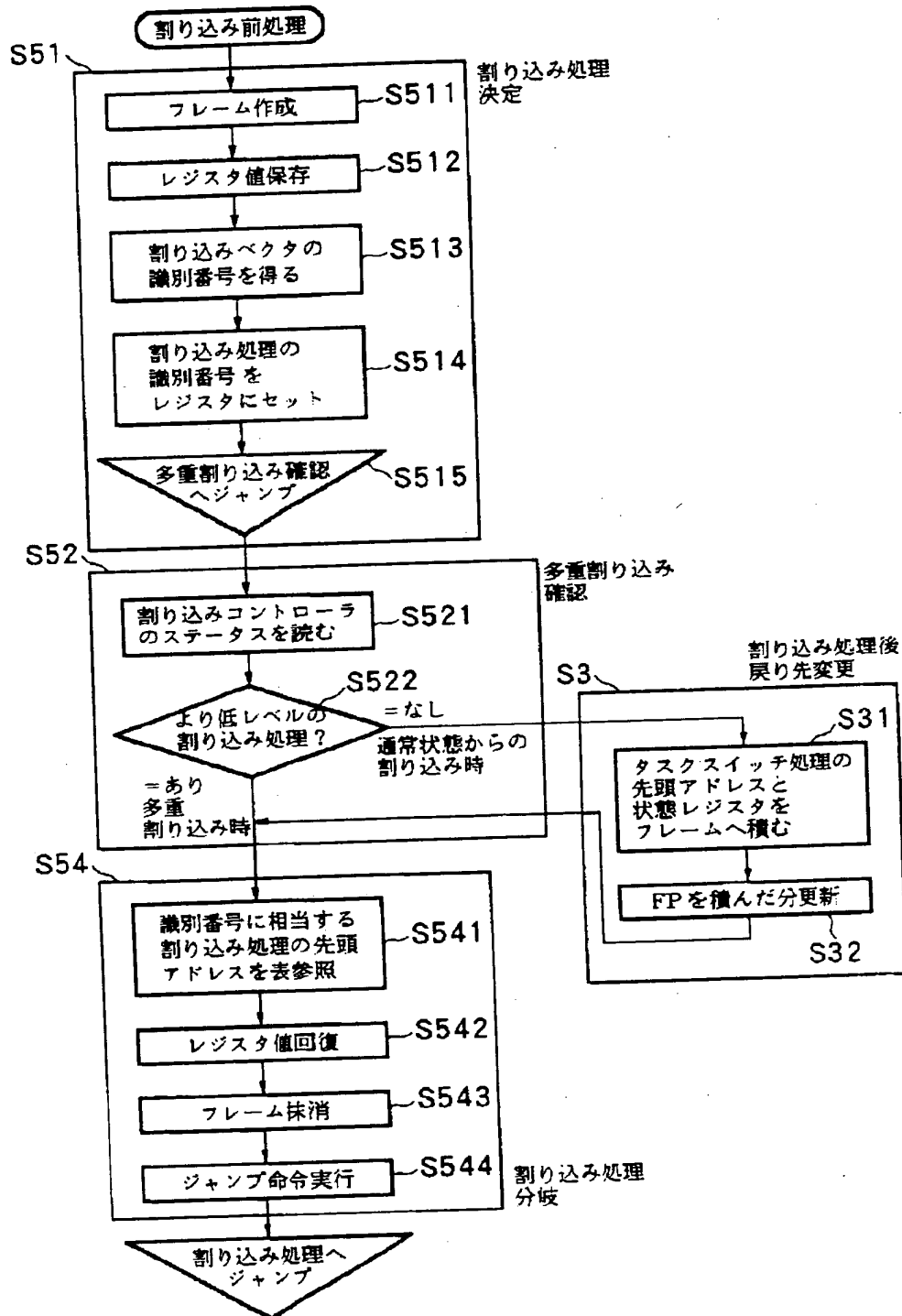
【図6】



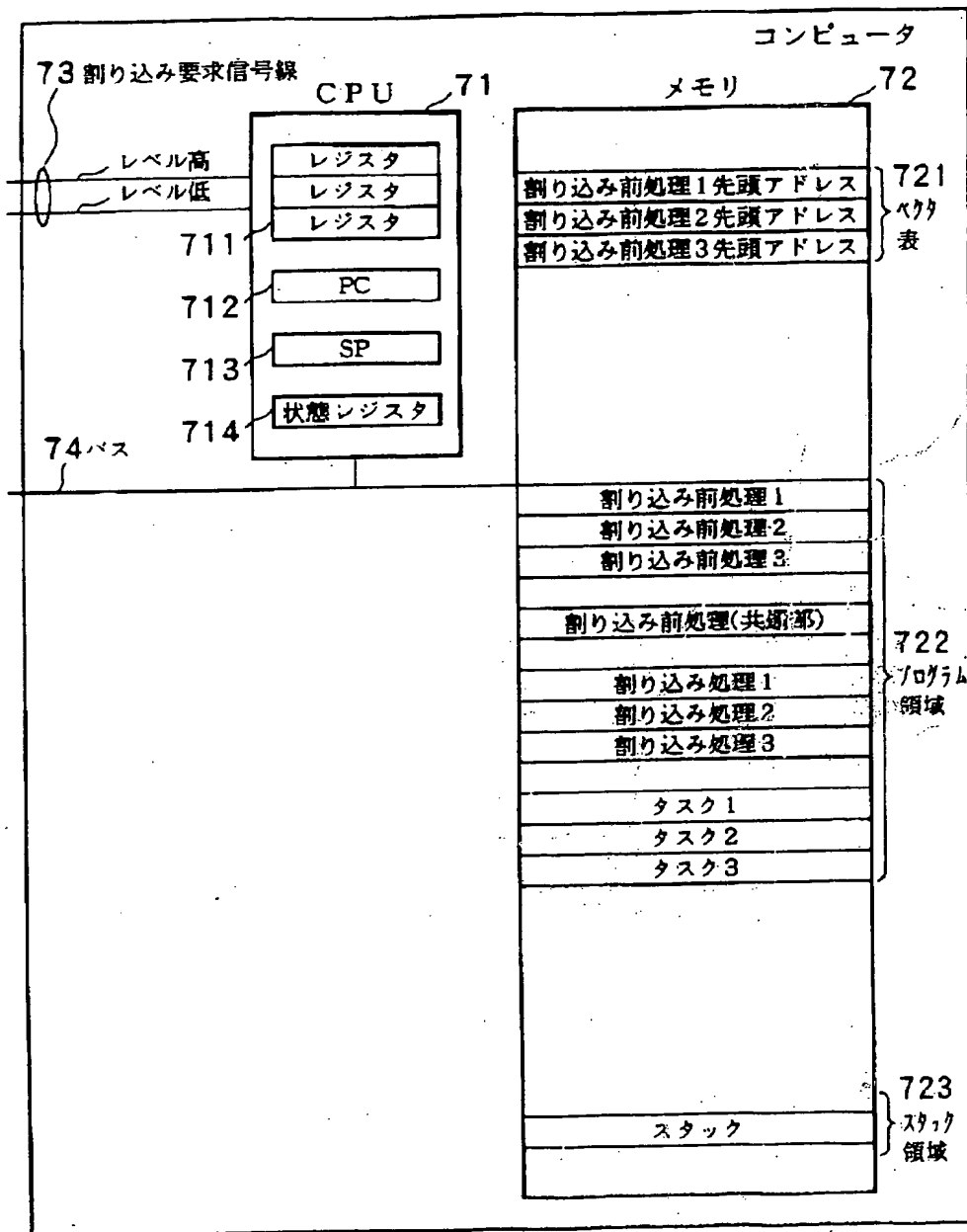
【図4】



【図5】



【図7】



フロントページの続き

(72)発明者 山田 潤二
 東京都大田区下丸子3丁目30番2号 キヤ
 ノン株式会社内

THIS PAGE BLANK (USPTO)